Sujan Pandey

Development of a Web Application (iDomain) in the CodeIgniter Framework

Hosting Accounts Management System Development

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis

9 May 2016

Helsinki Metropolia University of Applied Sciences

Author(s) Title Number of Pages Date	Sujan Pandey Development of a Web Application (iDomain) in the CodeIgniter Framework 31 pages + 7 appendices 9 May 2016
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Jaana Holvikivi, DSc.(tech)

The purpose of this thesis is to explore the scope of PHP in an action-based study and development of a web application for a company called Rockmandu Developers. The web application development procedure was carried out systematically in a hierarchical order to organize an order of tasks like system analysis, designing, development and testing. This project provides brief information about the developed web application and explains the web application development processes step by step.

Domain Management System (iDomain) developed under Rockmandu Developers is a system developed to meet the requirements of any web-hosting company who attempts to manage domain and hosting accounts of their customers. The PHP/MySQL platform in the CodeIgniter framework has been allocated for the initialization of the development process.

The Domain Management System (iDomain) is a web application that basically focuses on systematic recording and maintaining of information about domain and hosting accounts. It includes various functionalities like inserting records, updating records, deleting records, alerting about an expiration of domains and hosting accounts. It provides faster access to the required information so that work can be done in an efficient manner with accuracy.

Keywords	Web application, Php/MySQL, CodeIgniter Framework, WAF, ERD, DFD

Contents

Abbreviations and Terms

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
ASP	Active Server Pages
CD	Context Diagram
CFML	ColdFusion Markup Language
CSS	Cascading Style Sheets
DFD	Data Flow Diagram
DMS	Domain Management System
DOM	Document Object Model
ERD	Entity Relation Diagram
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
IT	Information Technology
JSP	JavaServer Pages
MVC	Model View Controller
MySQL	MY Structured Query Language
SDLC	System Development Life Cycle
SQL	Structured Query Language
UI	User Interface
UML	Unified Modeling Language
URL	Uniform Resource Locator
WAF	Web Application Framework
WWW	World Wide Web
XAMPP	X (cross platform), Apache, MySQL, PHP, Perl
XHTML	Extensible Hypertext Markup Language

1 Introduction

Information Technology as a field of information and communication technology has evolved exponentially throughout a decade. The development of the IT sector has a direct effect on the humankind and has brought positive changes into the daily lives of people. In comparison to other technological outputs, the IT out-turn has been quickly adapted by people. The significance of the IT sector development has been extended to every corner of the world and has benefitted billions of people. IT has rooted its section in every modern infrastructure of development and a drastic change has been brought in the communication related sector. The internet-based communication system is widely used nowadays for services such as instant messaging, emails and video conference. The internet is an interconnected form of electronic communication methods, and is also named as the Net or the World Wide Web.

The World Wide Web (WWW) is a systematic mechanism of a server which supports and operates hypertext formatted documents containing markup language called HyperText Markup Language (HTML). The HTML formatted files support links to other documents and also images, audio and video files, which are applicable for web browser. Web browser acts as a tool for accessing contents of the World Wide Web like websites and web applications which use an understandable language and script like HTML, CSS and JavaScript respectively. A web application has the functionality of submitting and retrieving data to or from databases and those data are generated dynamically through a web server. (Sciberras 2016)

This thesis scrutinizes web application development based on the idea to inaugurate a platform for a domain and hosting accounts management system, named as iDomain, for Rockmandu Developers, which is an IT related company located at Kathmandu, Nepal. The entire involvement during this thesis project is in the research and development of Domain and Hosting accounts Managing Web Application.

The general purpose of this thesis is to develop a web-based application using PHP/MySQL in the CodeIgniter framework. The project enhances abilities to work in the backend development as well as frontend development. It also provides knowledge of web-based application development and system design and development. The prob-

lem and challenges for developing an application is the time limitation. The development process has to be finalized within a short period of time.

2 Web Application Frameworks

The web application development project Domain and Hosting accounts Management System (iDomain) development task carried out under “Rockmandu Developers” company, is a system developed to meet the requirements of any web hosting company who attempts to manage domain accounts of their customers. Domain and Hosting accounts Management System (iDomain) is a web application that basically focuses on systematic recording and maintaining of information about domain and hosting accounts. It includes various functionalities like inserting records, updating records, deleting records, alerting about an expiration of domains and hosting accounts. It provides faster access to the required information so that work can be done in an efficient manner with accuracy.

In the initial phase, before starting a development process, the background study of platforms, tools and technologies is a very important part of any project. It helps the developer to provide updates of the latest technologies and other required information which could make tasks efficient. In addition, task-oriented studies also play an important role in conventionalizing the development process. The background studies were carried out in the topics explained in sections 2.1, 2.2, 2.3, 2.4 and 2.5.

2.1 Web Application

Web application is an evolved form of a website. The browser is a medium for retrieving and displaying web documents. The functionality of the exchange of data between the server and browser is the main feature of a web application. This mechanism was implemented to make an online-based system for practical use in people’s lives. Some of the current developed systems are in online shopping, social sites, banking services and online applications. (Pinto & Stuttard 2011)

A web application provides abundant information with the help of links it has for example, with intranet, extranet, libraries, database added to the system. Further, the program like iTunes player has the system where the application appears live on a desktop but music is retrieved from the web store. (Fowler & Stanwick 2004)

2.2 Web Technology Infrastructure

The tools and technologies that are used for the development of a web-related project are the infrastructure of the web technology. Currently, there are many tools and platforms available, which could be selected and implemented as per requirement.

Table 1: Basic Tools and platforms

Tools / Technologies	Details
HTML/HTML5/XHML	Markup language that most of the browser accept.
CSS/CSS3	Style sheet language
JavaScript	Scripting and programming language
MySql	Database
PHP	Server side HTML embedded scripting language
Ajax	Programming language that guides JavaScript to communicate with a web server.
Adobe Dreamweaver	Coding software platform
Adobe Photoshop	Graphics and Layout Design
Windows 2000 or above	Platform
Xamp/Wamp	Local hosting software
Apache	Web server application
Firefox, Google Chrome, Opera, Safari, etc.	Browser

Table 1 illustrats the basic tools and technologies that are used for development of a web application. For the operation of those tools, the system or hardware requirements are also an important part to be considered.

Table 2: Minimum Hardware and Software Requirements

Hardware	**Specification**
Processor	Pentium 3 or above
RAM	1 GB or above

Table 2 presents the minimum requirement of the hardware to run the tools and make workable platform for the development process. The modern system can easily fulfil the minimum requirement limitation.

2.3 Web Application Framework

Web Application Framework (WAF) is an organized system which provides a foundation or substructure for web application development. It also assists other web-related fields in customization, resource services and API-related tasks. The framework is a resource for the development process and makes the development and maintaining process easier and convenient and saves time and effort.

In a traditional PHP development environment, the developer is required to create interfaces for each aspect to the system, for instance an easy database access or user authentication. The framework is one kind of solution to this problem. In an application development environment, the reuse of code could play a vital role. The reuse enables efficiencies in the amount of new code required, which saves time and costs not only in coding section, but also in other tasks like design, coding and testing undertaken as part of the development life cycle. (Knupp 2014)

2.3.1 Library

A library can be defined as a collection of routines bundled together to provide functionality for a specific area. With the library, an application developer decides when and how to call a library function. There is no constraint, other than syntax, as to how or when an application developer decides to incorporate that library.

2.3.2 Inversion of Control

A framework provides a developer with an application skeleton into which a developer could drop their code and one could suggest that an application framework is a blank application waiting for the developer to fill in. Inversion of control means that a framework can take control of order processing, for example, with a call back. This concept is from the Windows GUI development environment such as JavaScript, where we add code to events, such as to click changes. The user or Windows decides when these events are executed.

2.3.3 Extensibility

Extensibility is another key feature of the framework. It is an ability to override or extend a base class using inheritance and add or replace features to provide new functionality, for example, a condition while overwriting a user identity cost to change the way that usual authentication is handled.

2.4 PHP Frameworks

PHP is a server-side scripting language which is widely used in building dynamic web applications. PHP as a hypertext pre-processor is very popular because of its flexible platform. For instance PHP can be embedded into an HTML source file instead of calling an external file for data processing. It is an open-source language and has various features related to the object oriented design. The use of PHP is still in good run. However, some of the modern alternatives for PHP could be CFML, Ruby on Rails, Perl, Go, Python with Django, Express.js, AngularJS, Node.js, ASP and JSP.

The PHP framework is a tool to well organize the development process of a web application written in PHP. It provides foundation for web application development. The development process becomes more efficient due to a built-in coding platform which can save time mainly in the coding section. The working procedure of the PHP framework is based on the Model View Controller (MVC). The MVC is a constructive design in the programming. It divides the development process into three sections, which will make it possible to concentrate and work on a particular field, without affecting others. This helps the building process in a simpler and time saving manner. (Reyes 2009)

Since this project is based on PHP, some of the popular PHP frameworks are presented below in figure 1.

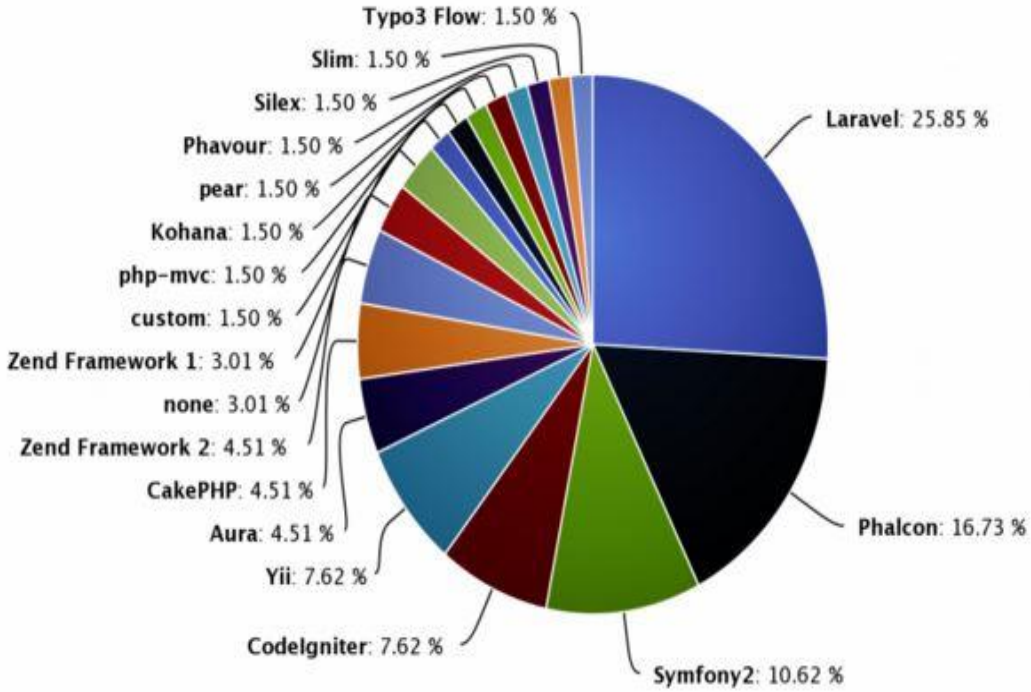


Figure1. Popular PHP Frameworks. (Rekovskaya 2014)

Figure1 illustrates the PHP frameworks which were mostly used and popular by the end of 2013. It reflects the Laravel as the most popular PHP framework with around 26% of popularity. Phalcon, Symfony2 and CodeIgniter follows second, third and fourth with popularity percentage of 16.73%, 10.62% and 7.62% respectively. (Rekovskaya 2013)

For this project the Codeigniter framework has been allocated. It is used during the development process.

2.5 CodeIgniter

CodeIgniter is a PHP-based open-source framework for the development of dynamic web applications. The development environment of CodeIgniter is very user-friendly and the core system uses a very low space for operation. Development of a web project by writing a code from scratch could be very time-consuming. CodeIgniter has a built-in library which contains sets of a pre-coded customizing structure and while in the development process, the developer can use it in their required field of a project to develop a system in an easy environment. The CodeIgniter platform is remarkable as it does not require any configuration to operate. The huge sets of libraries provide a lot of options for customization and freedom in using applicable coding rules. (Grace, Holscher & Leifer 2010)

2.5.1 Application Flux

Application flux in CodeIgniter is the process of data workflow around the system. It describes the algorithm and flow of a systematic chain relationship between layers.

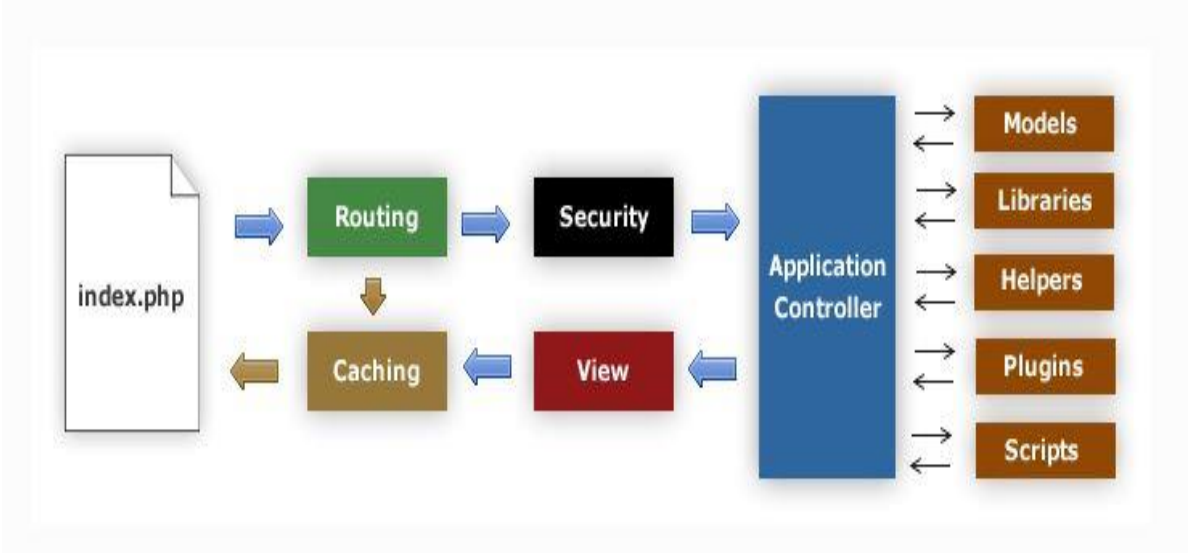


Figure 2: Application flow chart. (Grace, Holscher & Leifer 2010)

Figure 2 illustrates the application flow chart. It displays the procedure of a data flow. The initial and final stage of a data flow is handled by index.php and the web browser respectively. The index.php is a base section and it is the initial controlling medium. The information passing through index.php as HTTP requests is analyzed by the Routing section. Followed by Caching, it bypasses the system execution. Filtering the HTTP requests of submitted data is performed by the Security layer for final initialization through the controller section. The application controller initializes a recurred section like Plug-in, Scripts or Libraries, to access required request. The last phase is the browser section to view content. (Grace, Holscher & Leifer 2010)

3 Domain Management System

Domain Management System (iDomain) is a web application developed in Php/MySQL with the Codeigniter framework. It keeps records and alerts about expiration of all domains and hosting accounts opened at Rockmandu Developers Office. It is a generic application useful for any web hosting company. The major objective of this system is to keep record of domains in a systematic manner and retrieve the required records in a timely manner.

Interactive user interface makes this system easy to operate for both technical and non-technical users. iDomain is a web based system aimed to increase accuracy and efficiency in carrying out managerial tasks.

3.1 System Actors

The actor is basically an impetus of any system. The control system is handled by the actor and it plays an important role in allocating a sub-division of the role and control over its secondary actors. In this system, there are three types of actors that play the role of operating the system. They are listed below:

a) Super Administrator User

- Manage User
  - Add Users
  - Edit Users
  - Update Users
  - Delete Users
  - Reset password of users

- Manage Customer
  - Add Customers
  - Update Customers
  - Edit Customers
  - Delete Customers
- Manage Domains
  - Add Domains
  - Update Domains
  - Delete Domains
  - View details about Domains
  - View Normal Domains
  - View Expiring Domains
  - View Expired Domains
  - View Expiring Host
  - View Expired Host
- Update domain status

b) Administrator User

- Manage Domains
  - Update Domains
  - Delete Domains
  - View details about Domains

c) Normal User

- View details about Domains

3.2 System Development Procedure

The traditional System Development Life Cycle (SDLC) is used to develop the system. SDLC system is an approach of generating a statistics structure, imitation and mechanism which could be used by ordinary users to explore a system. It helps to demonstrate overall view of different phase of development procedure used in software development. The stages and their interconnection can be displayed in figure below:

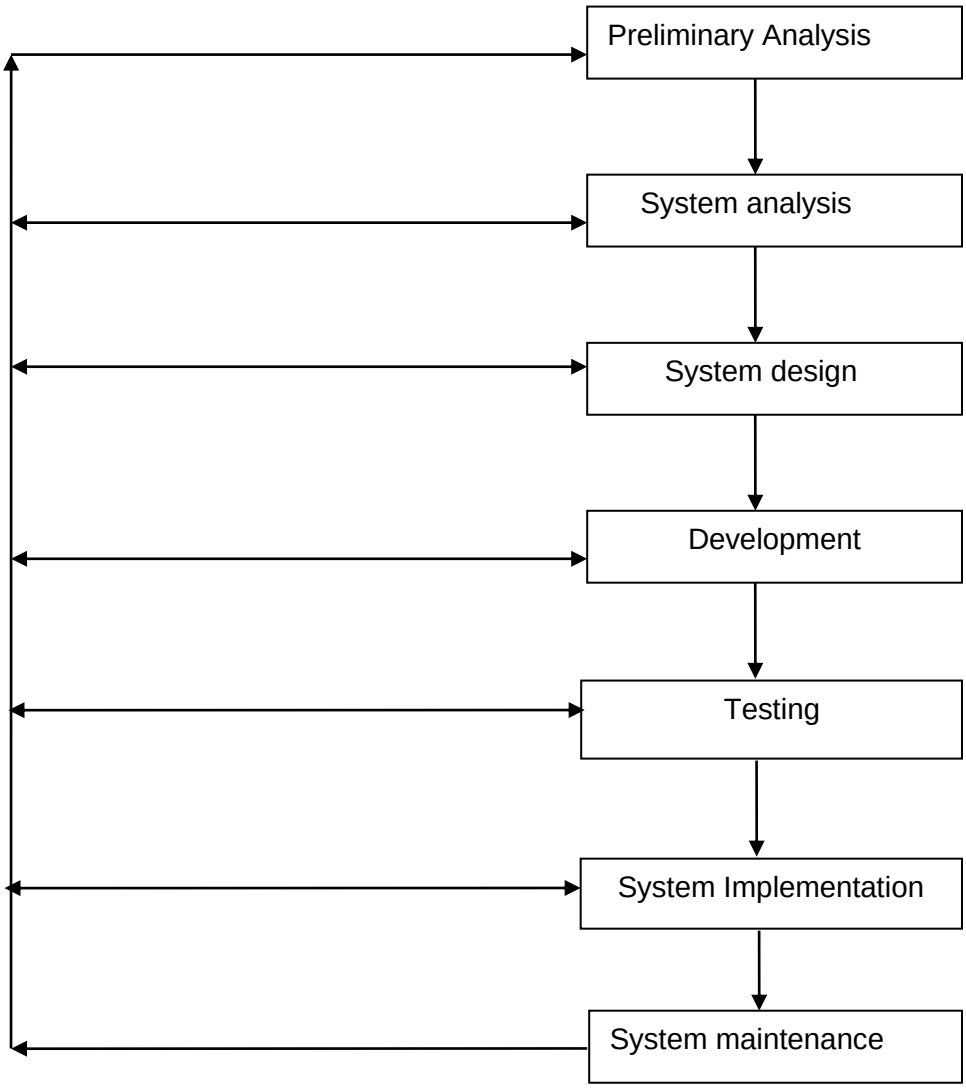


Figure 3: System Development life cycle. (Russell 2013).

Figure 3 shows the system development life cycle. These stages form a mechanism and framework for sketching a layout, making plans and implementation of planned stages of working order in development section.

3.2.1 Preliminary Analysis

Firstly, exploration and gathering of all data and information about iDomain was carried out. Then, analyzing the existing system and collected information via a questionnaire, observations, and an interview was done. After that, the project schedule was prepared for cost estimation and allocated resources for the SDLC. The company is new to the online business, so the company was not aware of what type of system they actually wanted to develop. So, it was difficult to convince the implement ideas and collect data and develop this system.

3.2.2 System Analysis

After analyzing the current existing system of the company, a feasibility analysis and risk analysis for a system development were performed. The company is currently using a manual system for their operations.

I) Viability Analysis

Viability analysis is the stage to check ability for the completion of a task. The initial planning and scheduling rough sketch helps to analyze the limitations that a project could have and choose the selected tools and components. The viability analysis can be sub categorized in following basics:

a. Technical viability

Technical aspects are mainly considered as a backbone for the system. The viability in technical factors is prioritized for obtaining features of currently available and easily includable tools and platforms for efficiency. Mainly this phase deals with a software and hardware section required for a system development.

b. Operational viability

Operational viability deals with a matter related to the after-development situation. The operation analysis calculates an average percentage of work done by a system and positive factors that add points to make efficient and systematic task completion. The operations or function of the new system easily fits the working environment and the user can adapt to the system with little training. Simple web forms and easy GUI makes it easy for the user to use the system, so it is operationally feasible to implement the web application.

c. Economic viability

Economic analysis was done to analyze the cost to be incurred in a web application development. It is analysis about whether a system will be economically feasible to be installed in an organization. The main target for financial viability estimation is to identify the good factors that could be added by the built system. The organization can afford to implement this system so it is economically feasible to develop this system. This system removes a manual system of the company so it is going to reduce some overheads.

II) Risk Analysis

The risk is a possibility of loss, damage, destruction or failure. Various possible problems were figured out during the development phase of a system along with their remedies. The possible risks are classified into four categories which are listed below.

a. People risk

- Parallel ongoing studies along with the web application development project for the thesis, which divided time allocated for a project and full attention could not be given to the project.
- Incomplete information collected for the project.

b. Technology risk

- Due to lack or resources there was difficulty in the development process.
- Difficulty in working with development tools.

c. Tools risk

- Due to the lack of proficiency it was difficult to use tools for the development of the system.

3.2.3 System Design

After performing system analysis, a tentative framework of iDomain was designed. A use-case diagram, entity-relationship diagram, and context diagram were developed for further processing. The design of a data flow diagram with suitable input, process, storage and output was preceded.

3.2.4 Development

After accomplishing system design, the development process was started. The rough sketch of a module built in Adobe Photoshop was used as a reference for designing the layout. The XAMPP program was installed in a Windows system to work locally. After setting up a local hosting environment, the CodeIgniter framework was set up in the htdocs file in the XAMPP program. The application folder of CodeIgniter is the mostly used part for a web application development process and the entire task performed during a process is stored there.

The coding of the system was done using Dreamweaver CS6. The FTP file transfer to localhost server was setup in Dreamweaver. The setup was done in Dreamweaver by adding the server information for site section of Dreamweaver. For server setup, a connection using the local network has to be selected and the server folder has to be redirected to the CodeIgniter folder in the htdocs folder of the XAMPP program. The web URL begins with http://localhost/ and it follows final destination as a directory to the application folder of CodeIgniter in the htdocs folder. The connection between the

Dreamweaver and CodeIgniter framework in the localhost is set up to run a program and get a live view of coding outcome. The XAMPP program should be turned on during the entire process.

Further, the coding and development process was preceded according to the designed architecture system and layout. The individual parts are designed and developed separately, and after finalizing each section's requirement, in the final stage all those parts were assembled and built as one working system.

3.2.5 System Check

To examine the output of iDomain, system testing was performed. The tests of the output and efficiency of a developed application were also performed. The checks are categorized in the following subsections.

I) Module

A module check is done when the finishing of a module is completed. An individual module has been tested and verified before integrating them. After the module are tested and well verified an integration of the modules is done. Module testing has been conducted to ensure the code written for each section is in the functioning order. Two labeled box checks, white and black box checks are done in this phase. The white box check includes the checking of the internal structure of the application including coding section, database tables, framework components, etc. and the black box check includes the checking of the functional part or the outer structure of an application, for instance checking of the inputs and outputs methods of the registration form. After this phase, the system is tested again to check an output.

II) Functional operation

A functional operation checks are done when the function of a system is built to confirm the planned operating function of a system. The functionality check is an important section which could be a base to analyze system concession and its needs. In this phase

the black box check is done and to perform a task there is the requirement of understanding the central idea of cognition.

III) Interface

Interface testing is done to identify communication between servers and their functioning sections. The error messages are important to locate the problems and they are corrected as needed. Whenever the system displays a fault message, the browser should catch and display the appropriate error messages.

3.2.6 System Implementation and Maintenance

After the developed system was tested and verified, the system iDomain was implemented and made ready to go on a live server. As the company was not using any web based system, the replacement of the existing system was not made. So, direct implementation of the system was done.

The final stage was based on the maintenance of the iDomain. In this phase, the system was reviewed regularly and its performance was tracked. The web application development time period was very short and after implementation there was some error in the application. A PHP error was encountered displaying the message “only variable references should be returned” in the common.php file. Following the system maintenance procedure an error was identified as a version error and fixed within a short period of time. In this way maintenance activities were implemented.

3.2.7 Limitations of the System

This application is only a small step that has been taken at developing a management system. The time limit and limitations of the working platform selection abandoned the plan of integrating a management system that could also handle the maintenance of overall tasks. This application is very premature and at its tender age. The major task

ahead would be to extend the features and capabilities of iDomain so that it would be able to compete with the existing applications in this field.

4 Architectural Overview

The system is Domain and Hosting accounts Management System which is a web-based system. There are two server components. These server components can exist on same server computer or can be different computers for each such as a database server or web server.

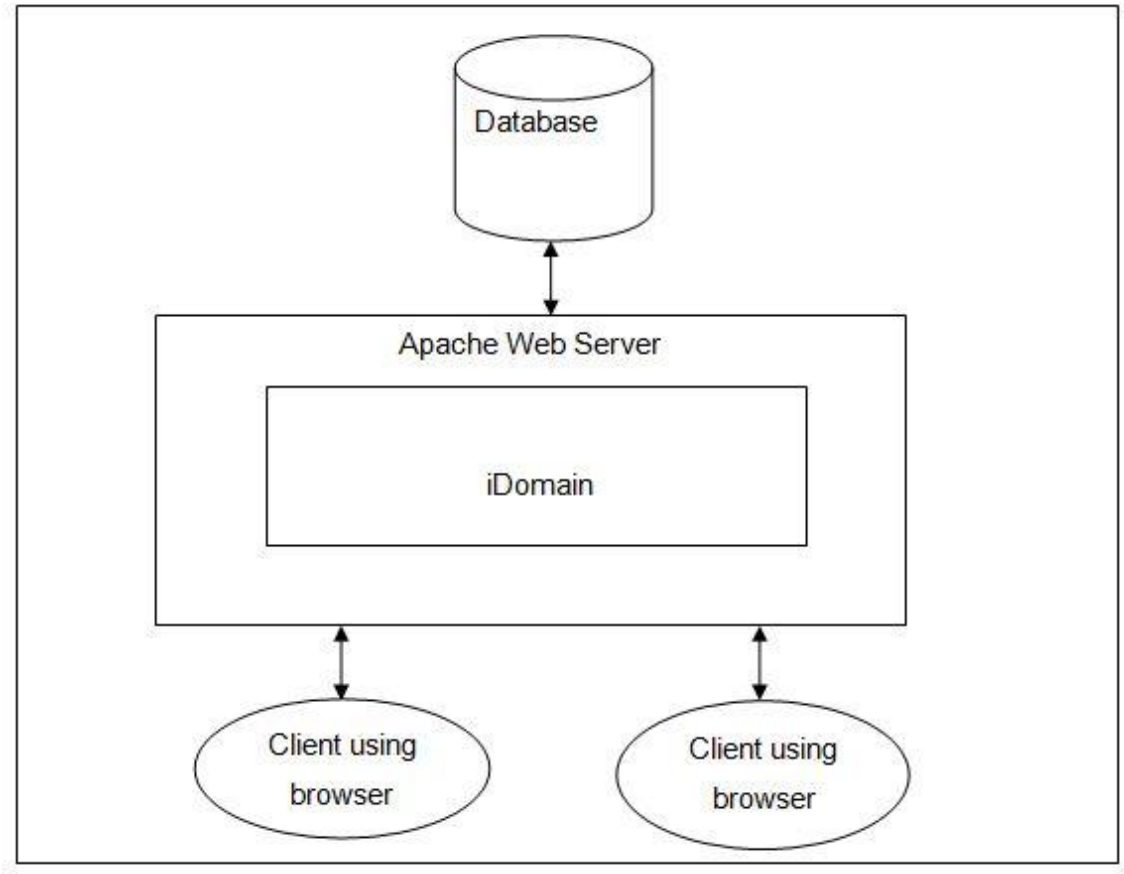


Figure 4: Architectural Overview of Domain Management System. (Gemmell 2007)

As displayed in figure 4, the Apache web server will communicate with the database server and provides the data that are required for the application iDomain. The application for the clients is a web browser.

4.1 Component Architecture

The Domain Management System application has three layers which are described below.

4.1.1 Presentation Layer

The presentation layer is responsible for presenting data and information to users in the format that users can identify with, i.e. in a high level language. It is also liable for collecting events, responses and data from various users. The presentation layer responds to a user request by receiving the required result from the application layer and presenting in a predetermined format. (Tarhini 2011)

4.1.2 Application Layer

The application layer acts as a mediator between the initial and final layer. It receives the input from the presentation layer and is also responsible for creating pages, validation and application management. It gets data from users, validates the data, prepares the data as per the requirement and generates appropriate messages to users. Then it will communicate with the data access layer for further processing. It can also send data to the presentation layer and if any of the user's actions and data do not confirm to the business logic of the system, then it will be valid and a necessary response will be provided to the presentation layer. (Tarhini 2011)

4.1.3 Data Layer

The data layer receives information from the application layer about the selected requirements to be displayed. It initializes application layer to identify information from a database and formulate an action to pass the information to next layer. Data layer processes that information as requested and if required, communicates with database for further processing. The result received from the database is again presented to the application layer. (Tarhini, 2011)

4.2 System Design

The system design or unified modelling language (UML) has been used since 1997. Then main objective of this language and system is to provide a standard and common platform to develop a computer application. UML provides a standard convention of graphics to visualize all models, which would provide well-defined semantics behind each design patterns. UML specifies models precisely, unambiguously and completely. In addition, it also provides specification for all artefacts like analysis, design and implementation decisions. It is not a real coding language but many of its models are used to illustrate the design pattern of the popular programming language like JAVA, C++ and PHP. All of the details of the design pattern are mapped by keeping a well-organized documentation. Executables are addressed by the UML along with the system architecture. (Fakhrautdinov 2009). The following UML diagrams are used for this project.

4.2.1 Use-Case Diagram

A use-case presents the overall functioning order of the system. It provides developers with a rough view of how functionality is divided and guides on to implement the required role on it. The diagram highlights the administrator of the system and allocates the additional user space to distribute the functionality. (Fakhrautdinov 2009)

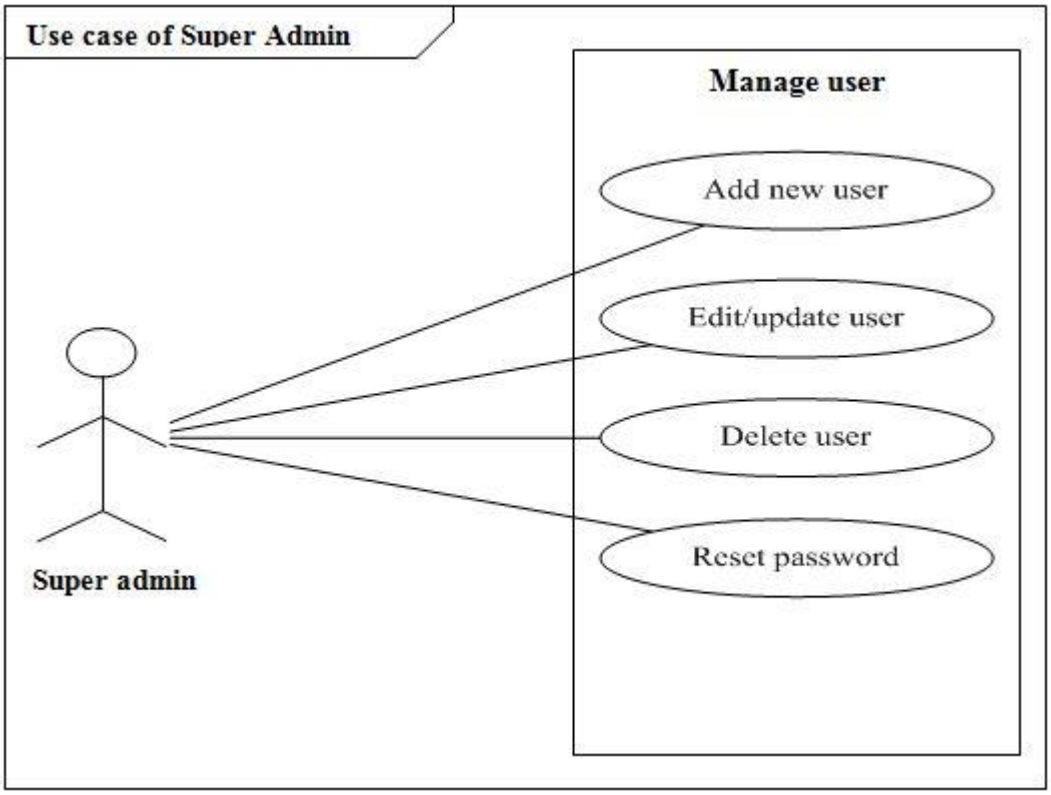


Figure 5: Use Case View of Super Administrator to Manage Users

Figure 5 depicts the use case diagram of an actor Super admin that has the power to manage users. In use case view of Super Administrator to manage users section, the Super admin can add new users, edit/update existing users, delete existing users and reset the passwords of users.

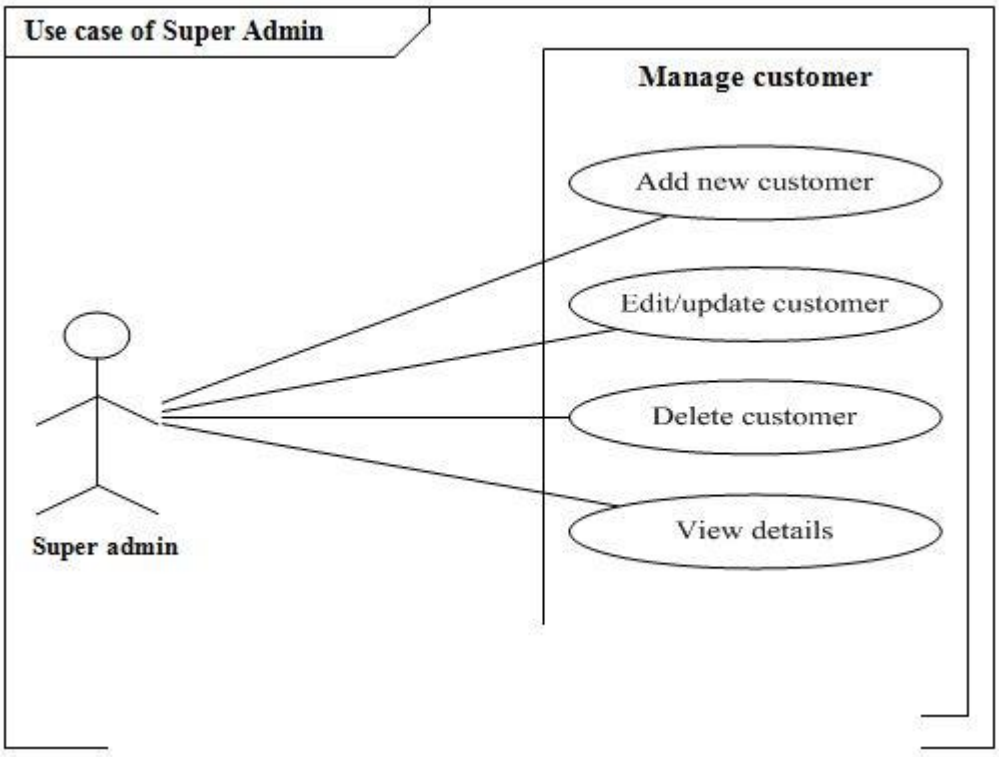


Figure 6: Use Case view of Super Administrator to Manage customer

Figure 6 depicts the use case diagram of an actor Super admin that has the power to manage customers. In use case view to manage customer section, Super admin can add new customers, edit/update existing customers, delete existing customers and view information of all customers like their name, location, mobile numbers, office number and email address.

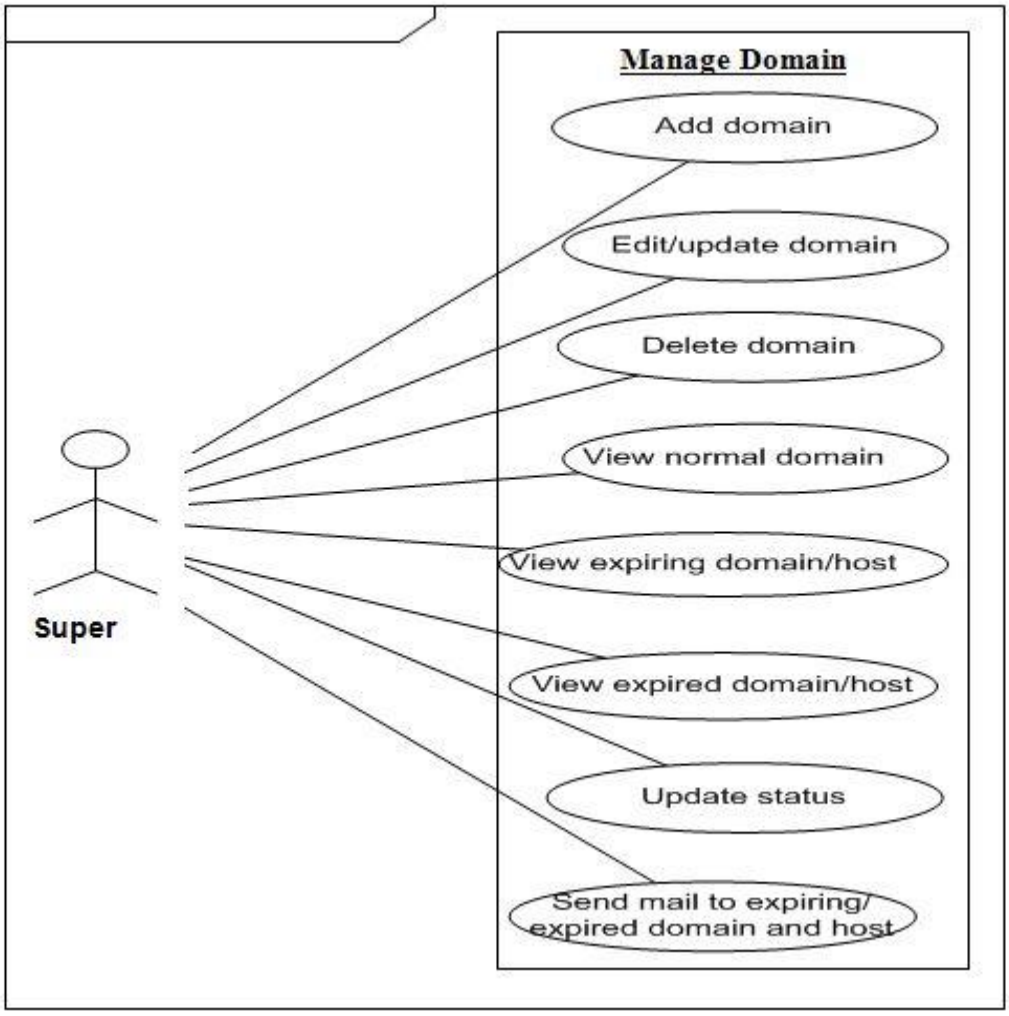


Figure 7: Use Case view of Super Administrator to Manage Domain

Figure 7 depicts the use case diagram of an actor Super admin that has power to manage domains. In use case view to manage domain section, Super admin can add a new domain, edit/update existing domain, delete existing domain, view normal, expiring and expired domain and hosting account, send mail to customers having expiring and expired domains and hosting account and update the status of the domain and hosting account. The Super admin can view every detail of the domain and hosting an account like domain name, domain registered date, domain expire date, hosting provider, hosting registered date, hosting expire date, domain and host status, name server and hosting space.

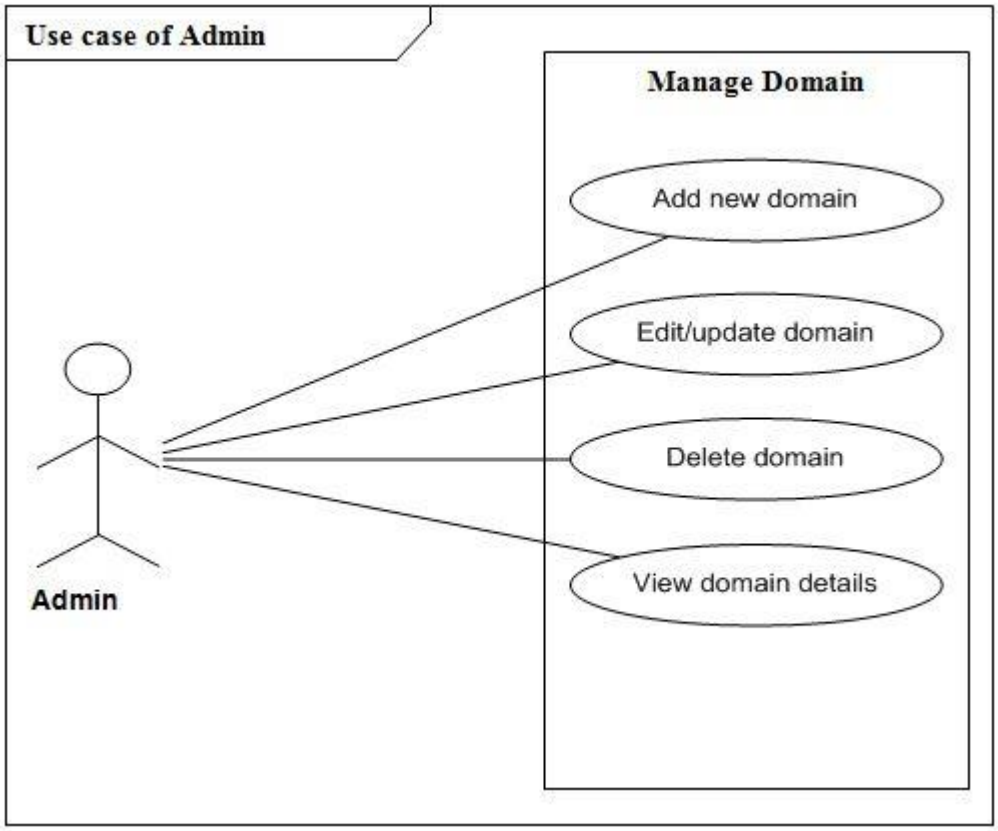


Figure 8: Use Case view of Administrator

Figure 8 depicts the use case diagram of an actor Administrator that has power to manage domains. In use case view of Administrator section, Super admin can add new domain, edit/update existing domain, delete existing domain and view details of all domains and hosting account like domain name, domain registered date, domain expiry date, hosting provider, hosting registered date, hosting expire date, domain and host status, name server and hosting space.

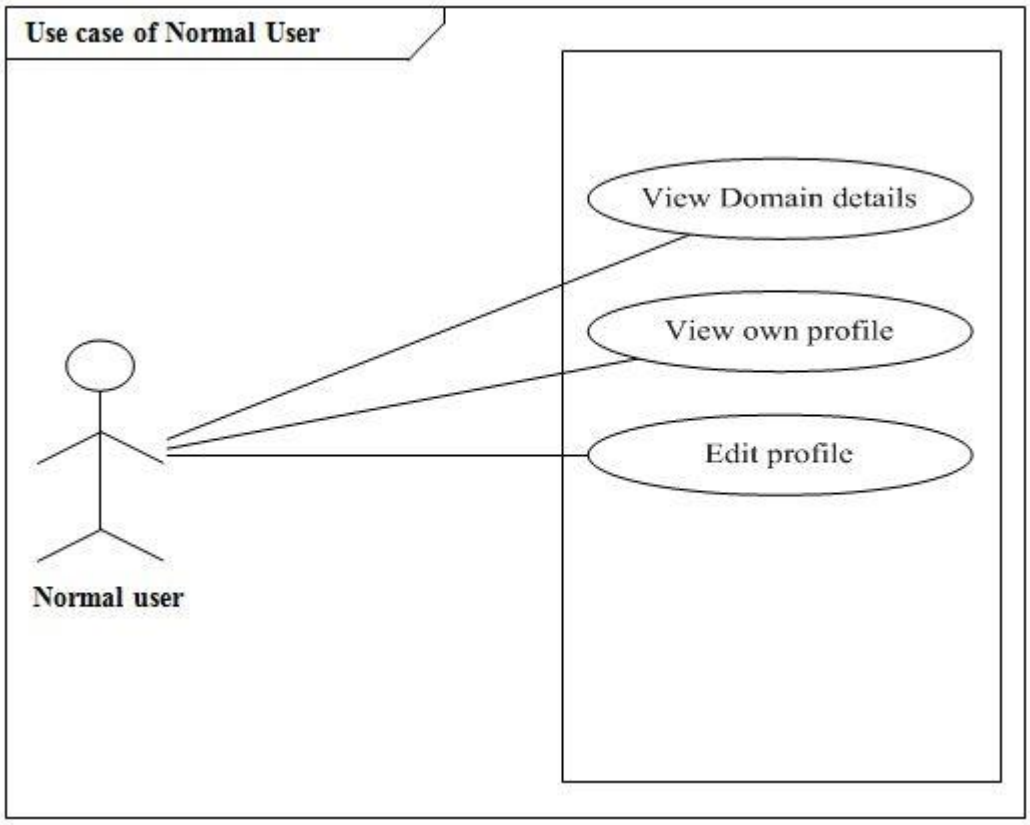


Figure 9: Use Case view of Normal User

Figure 9 depicts the use case diagram of an actor Normal. In use case view of normal user section, the normal user can only view details about the domain. Normal user also can view and edit their own profiles.

4.2.2 ER Diagram

Entity-relation diagram (ERD) is a high standard data modeling system, which assists developers to innovate imaginary models and graphical representations of database design. It provides an overview of a developed system and how data are represented and organized in the different sections of the final database. ER diagram modeling system was introduced by Professor Peter Chen. It is a tool or platform which could mediate between designer and users. (Dybka 2015)

The presentation of organized data into an analytical layout and diagram provides a larger view of the mechanism. The diagram has its own components and symbols for representation of functions. They are: Entity, Attribute and Relation. These components are implemented for the construction of a full module and are presented in figure 10.

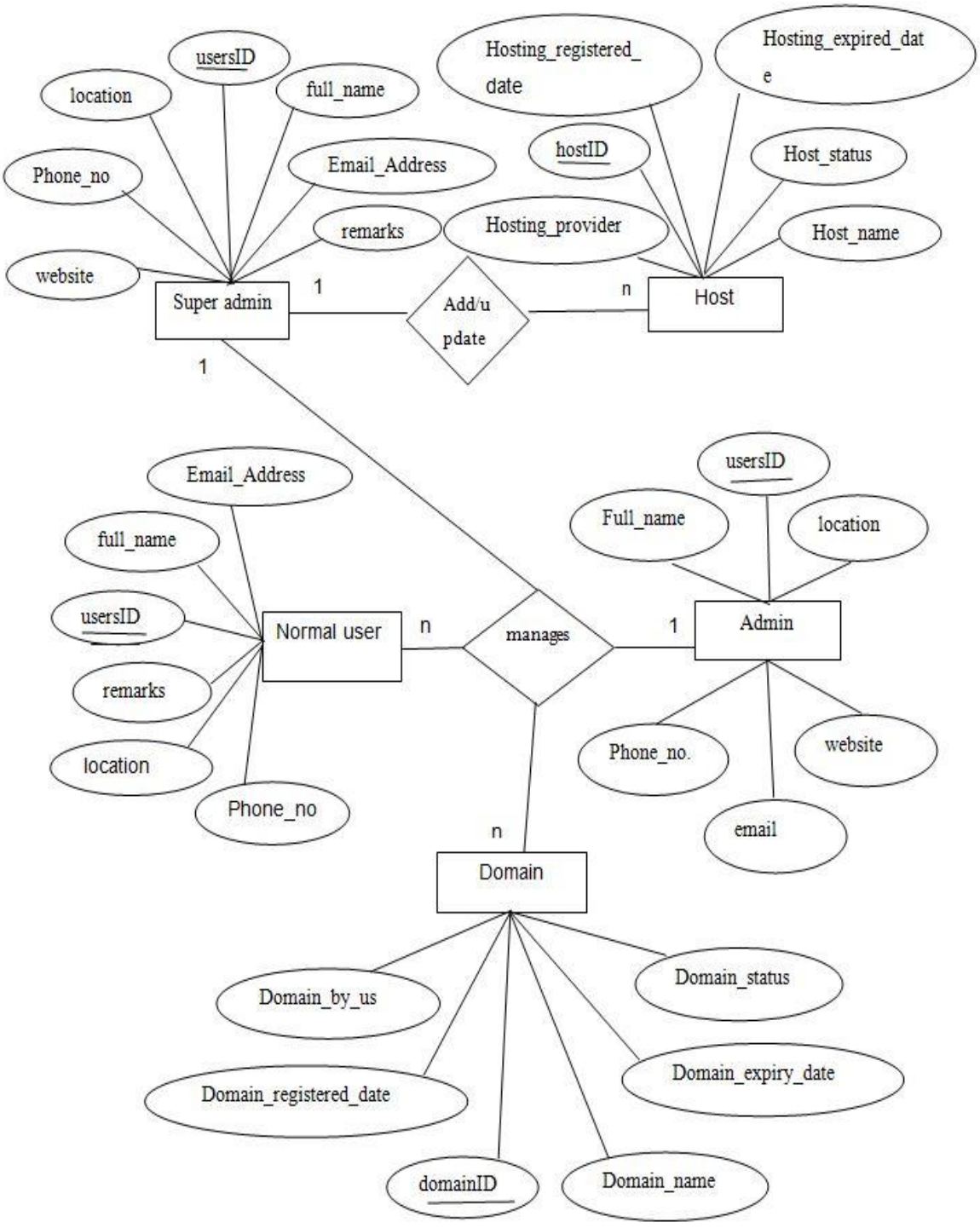


Figure 10: Entity Relationship Diagram

Figure 10 shows the Entity Relation Diagram that provides an overview of the developed system and how the data are represented and organized in the different sections of the final database. The component of the ER Diagram is organized according to function allocation of the developing database system. The entity classes and attributes are presented by shape of rectangles and ovals, respectively. The entity classes and attributes get connected through solid lines whereas the underlined attributes are key attributes. Multi-valued attributes and derived attributes are normally represented by double ovals and dotted ovals, respectively. However, those attributes are not in use in figure 10. The diamond shape attribute represent actions and it reflects a relation between two entities and how they share information in a database. The number “1” and alphabet “n” are cardinality. Cardinality is used instead of arrowheads. As in figure 10, one admin can manage many user and domain, so a cardinality ratio is written as 1:n. This is one to many relationships. Finally, after completion of the ER Diagram design the database will be created. ER diagram acts as the framework for the designing and displaying mechanism of the database. (Dybka 2015).

4.2.3 Context Diagram

A context diagram (CD) is a simpler way of sketching the modelling of a function of the system within a small layout. It helps developers to introduce a high-level model of an available or used system which could identify the limitations and interests of the system to respond with other functions on existing platforms. A Context diagram is a layout which could present the central interest of the system in a single picture. However the interior design is not included and all functional parts which are parts of interaction are set to be part of the environment of the system. (Castiglioni & Losacco 2009)

.

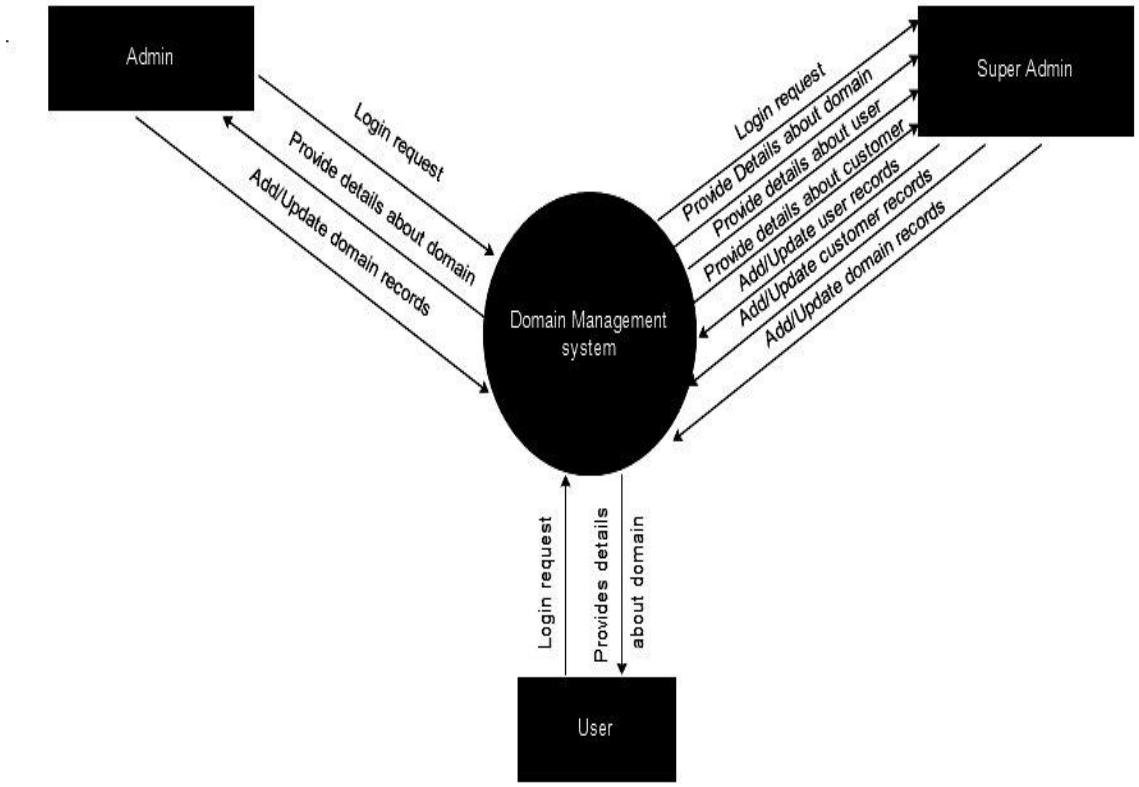


Figure 11: Context level Diagram. (Castiglioni & Losacco 2009)

Figure 11 illustrates the high level view of the planned environment of the context-level diagram. It displays the environment under consideration of the developing system. Domain and Hosting accounts managing the order of process and flow of information from one phase to another conveys an exchange of data and interconnects to the relationship of the system with outer factors, i.e. Super admin, Admin and Normal User.

The Super admin requests to log to the system and enters the system after confirmation of the security codes. The system provides super admin with details of domains, customers and existing users. The Super admin has a power to add, edit, delete and update domains, customers and users at any time. Similarly the Admin also requests to log in and in return the system provides the admin with details of only the domain and power to add, delete and edit existing domains and hosting accounts. The Normal user also requests to log in and in return the system provides the normal user with details of the existing domains. The Normal user does not have any authority to add/update any domains.

4.2.4 Data Flow Diagram

DFD displays the circulation of information in the graphical chain system. The thumbnail of the system is created in the initial section, then preceded later in a complex system. It is a technical representation of modelling with a superior layout and information included presenting a process of transformation of the final result from input data in a series. (Donald & Vie 2000)

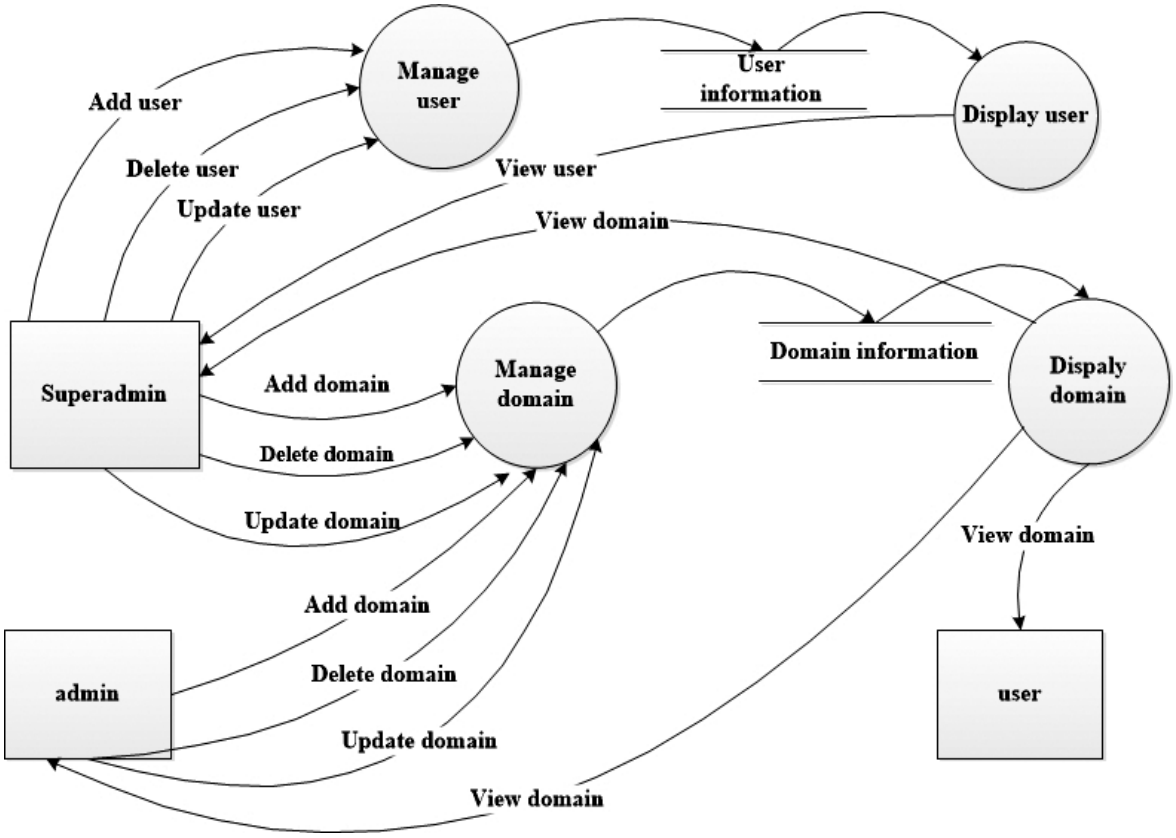


Figure 12: Data flow diagram. (Donald & Vie 2000)

Figure 12 displays the normal mechanism of data loading and output generation from the system. It shows that the procedure and stepwise techniques of how data is received from the initial phase and where it could be stored. Here the super admin has power to manage user by adding new users, editing or updating information about existing new users, deleting existing users. The output is then stored in the user information database. The information stored in the user database can then be viewed by the super admin user. Similarly the super admin also has power to manage domains and hosting accounts by adding new domains, editing/updating existing domains, delet-

ing existing domains and hosting accounts. The output is then stored in the domain information database. The information stored in the domain database can be viewed by the super admin.

Likewise, the admin also has power to manage domains and hosting accounts. The admin can perform the tasks like adding new domains, editing/updating existing domains, deleting existing domains and hosting accounts. The output is then stored in the domain information database. The information stored in the domain database can then be viewed by the Super admin and normal users.

5 Result

The development process of the iDomain web application was finalized. The developed system is the final result of the defined task. All the requirements that have to be included in the system were fulfilled. The developed interface of the system is presented below and in appendix 5, 6 and 7.

I) Interface of the developed system

Figure 13 shows the front section of the Super Admin page of the web application. This page is displayed after login using the Super Admin credentials.

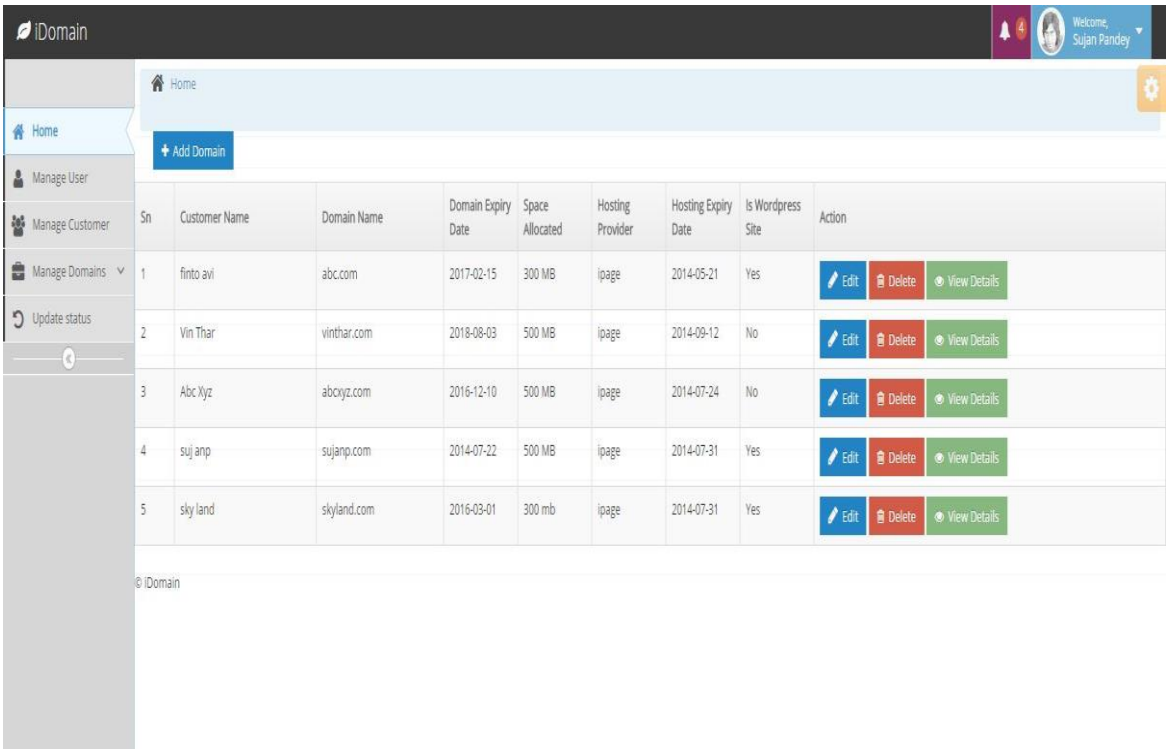


Figure 13: Home page of super admin

Likewise, the Administrator and the Normal User also have separate login pages and they could use the application as per the rights allocated by the Super Admin. The snipped images of different functions and user backgrounds are presented in appendix 5, 6 and 7. The functionalities of the web application which were expected to be implemented before the development process started were successfully finalized as per system design.

6 Conclusion

The main purpose of this thesis was to develop a web application for the Rockmandu Developers Company and explore information about the PHP platform in web application development. Two sets of goals were ranked during the process of development. In the beginning phase, surface and background study was carried out of the PHP-related field. In addition, the alternative modern medium for developing a dynamic website was also studied to know the latest updates of tools available and to choose the favourable platform to begin the development process.

Secondly, after an analysis of the requirements for the initialization of the development process the general architecture for an application was designed. Further, the task-based seven level of hierarchy was fixed for implementation of the entire required task for the development of the web application. Stepwise the procedure was followed from a preliminary analysis to the development process to the final system maintenance.

Finally, the initial goals targeted for the finalization of the project were achieved successfully. The project for development of the domain and hosting accounts managing web applications for Rockmandu Developers Company using PHP/MySQL in CodeIgniter Framework was completed.

References

Castiglioni Fabio, Losacco Vito. (2009). The Operational Context Diagram. Available at: https://www.ibm.com/developerworks/architecture/library/ar-introocdv1/#0.2.Introduction|outline
Accessed on: 31.3.2016

Donald S. Le Vie, Jr. (2016). Understanding Data Flow Diagram. Available at: http://ratandon.mysite.syr.edu/cis453/notes/DFD_over_Flowcharts.pdf
Accessed on 31.3.2016

Dybka Patrycja. (2015). ERD Notations in Data Modeling. Part 2. Available at: http://www.vertabelo.com/blog/technical-articles/chen-erd-notation
Accessed on: 31.3.2016

Fakhrautdinov Kirill. (2009). The Unified Modeling Language. Available at: http://www.uml-diagrams.org/
Accessed on: 31.3.2016

Fowler, Stanwick, Susan, Victor. (2004). Web Application Design Handbook. Sansome Street, San Francisco: Morgan Kaufmann, p.7
Accessed on: 31.3.2016

Gemmell Richard. (2007). AppThena: Architecture. Available at: http://www.appthena.com/developer/manual/architecture.html
Accessed on: 31.3.2016

Grace Bobby, Holscher Eric, Leifer Charles. (2010). Application Flow Chart: CodeIgniter Overview: Docs.
Available at:
http://www.codeigniter.com/user_guide/overview/appflow.html
Accessed on: 31.3.2016

Grace Bobby, Holscher Eric, Leifer Charles. (2010). Welcome to CodeIgniter: Docs.
Available at:
http://www.codeigniter.com/user_guide/general/welcome.html
Accessed on: 31.3.2016

Knupp Jeff. (2014). What is a Web Framework? Available at: https://jeffknupp.com/blog/2014/03/03/what-is-a-web-framework/
Accessed on: 31.3.2016

Pinto Marcus, Stuttard Dafydd. (2016). The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws. Crosspoint Boulevard Indianapolis, IN 46256: John Wiley & Sons, Inc., p.39
Accessed on: 31.3.2016

Rekovskaya Olga. (2014). PHP Framework for eCommerce Website Development. Available at:
http://www.issart.com/blog/php-framework-ecommerce-website-development/
Accessed on: 31.3.2016

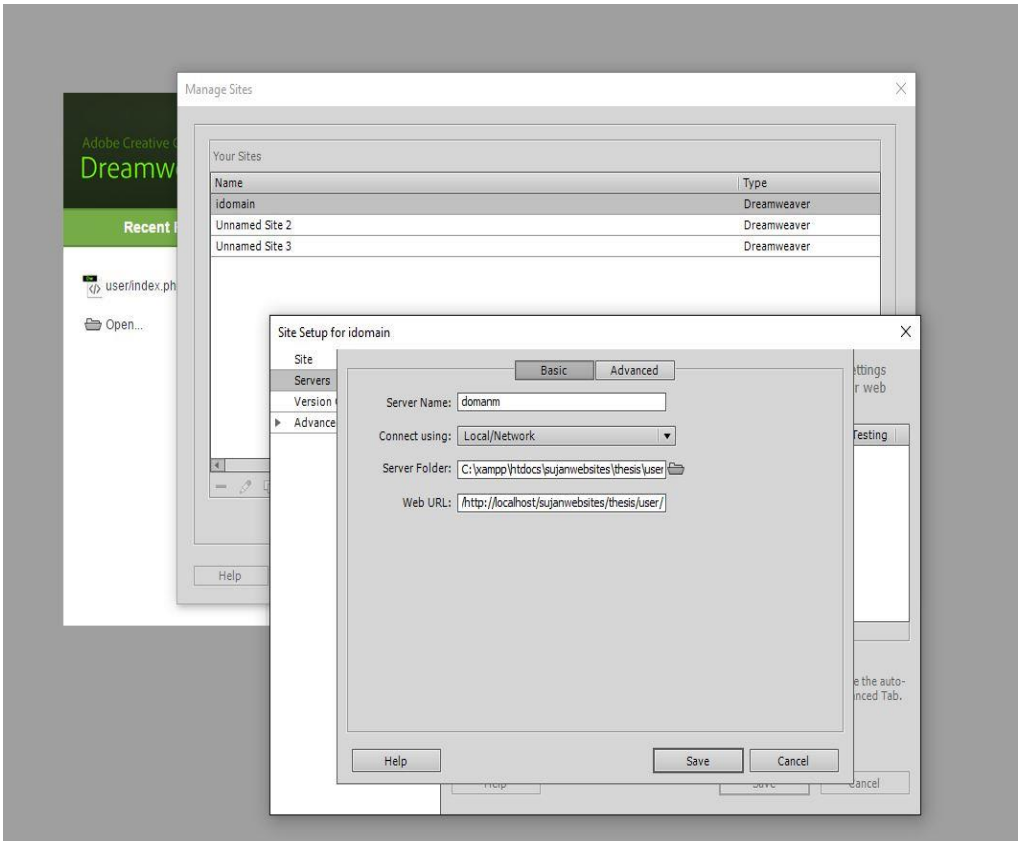
Reyes Joel. (2009). Discussing PHP Frameworks: What, When, Why and Which? Available at:
http://www.noupe.com/development/discussing-php-frameworks.html
Accessed on: 31.3.2016

Russell Mike. (2013). Understanding the SDLC isn't Just for Programmers. Available at:
http://execprojectmanager.com/understanding-the-sdlc-isnt-just-for-programmers/
Accessed on: 31.3.2016

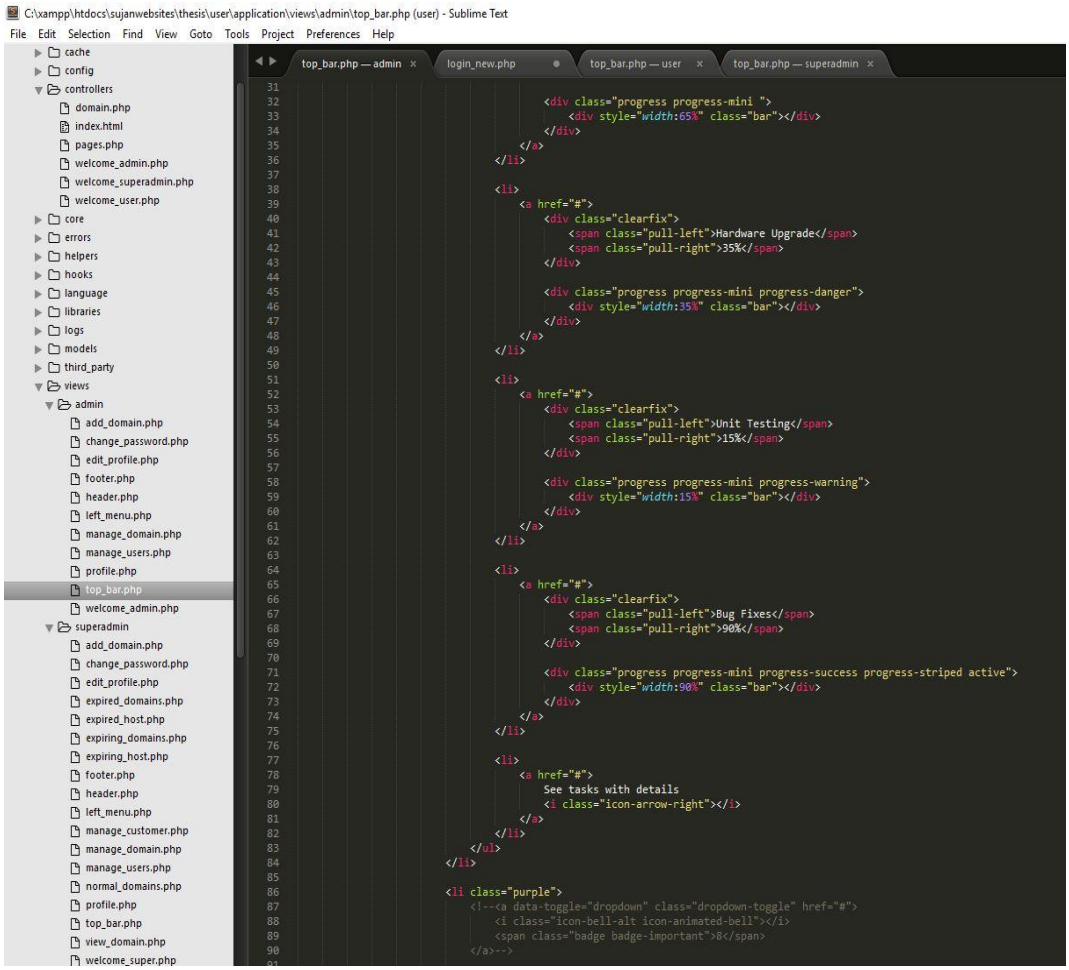
Sciberras Nicholas, (2016). Web Application: What are they? What of them. Available at:
http://www.acunetix.com/websitesecurity/web-applications
Accessed on: 31.3.2016

Tarhini Ali. (2011). Concepts of Three – Tier Architecture. Available at:
https://alitarhini.wordpress.com/2011/01/22/concepts-of-three-tier-architecture/
Accessed on: 31.3.2016

Appendix 1. Sites setting in Dreamweaver



Appendix 2. Controller and View panel in CodeIgniter



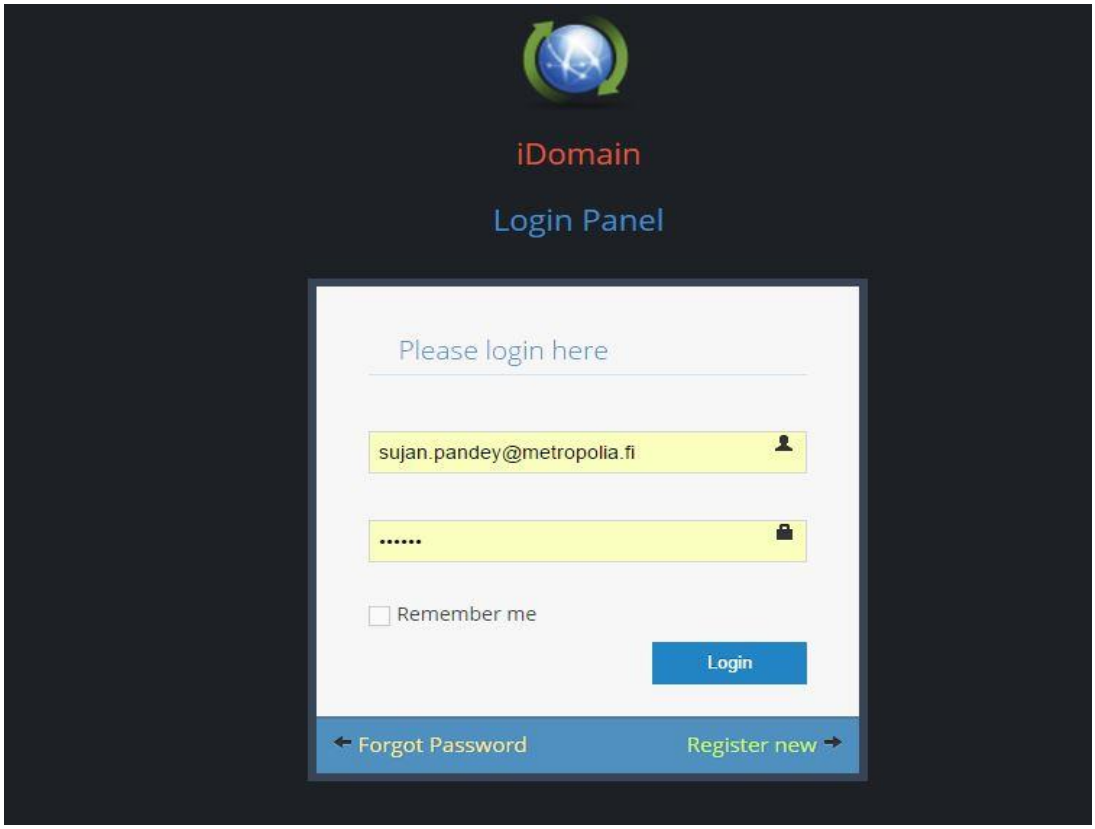
Appendix 3. User login page coding part snip

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>

<body>
<!--<form name="login_form" action="">-->
<?=form_open('domain/process_login')?>
<?php echo $this->session->flashdata('error'); ?>
<?php echo $this->session->flashdata('congratulation'); ?>
<br />
    Please Login <br />
    Email address: <input type="email" name="email_address" placeholder="email address" required="required"  /><br />
    Password: <?php
        $data=array(
            'name'=>'password',
            'placeholder'=>'Type your password here'
        );
    echo form_password($data); ?><br />
    <!--<input type="submit" value="Login" />-->
    <?=form_submit('submit', 'Submit')?>

<?=form_close()?>

</body>
</html>
```

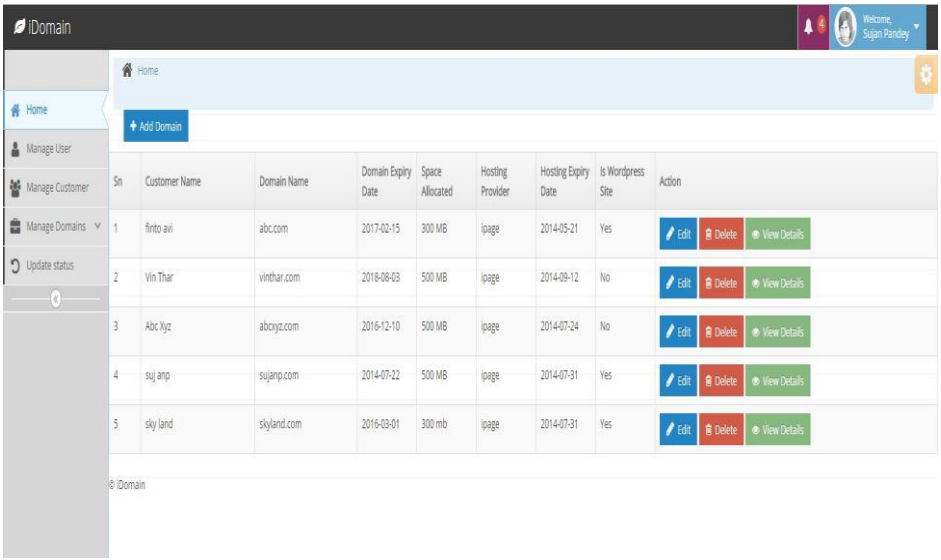
Appendix 4. Super Admin Add Domain Coding Snip

```php
<?php
$attributes = array('class' => 'form-horizontal');
?>
<?php
if($mode=="edit"){
     echo form_open('welcome_superadmin/update_domain', $attributes);
     }else if($mode=="view")
     {
        echo form_open('welcome_superadmin/view_domain',$attributes);
     }else{
     echo form_open('welcome_superadmin/add_domain',$attributes);
     }
?>
<div class="controls">
<?php if($mode=="edit")
     {
         echo form_hidden('domain_id', $domain_info->domain_id);
     }
     ?>
</div>
<div class="controls">
Customer Name:
<?php
if($mode=="edit")
{

        $data=array(
        'name'=>'customer_name',
        'placeholder'=>'Customer name',
        'value'=>$domain_info->customer_name,
                );
                echo form_input($data);


} else if($mode=="view")
{
    $data=array(
    'value'=>$domain_list->customer_name,
    'readonly'=>'readonly'
    );
    echo form_input($data);

}else {

    //echo '<select name="customer_name">';
   //echo '<option value="'.$customer_info->customer_name.'">'.$customer_info->customer_name.'</option>';
    //echo '</select>';
```

Appendix 5. Login Interface of Developed Web Application



Appendix 6. Super Admin Home Page



Appendix 7. Add Domain Section